REMARKS:

In the outstanding Office Action, the Examiner rejected claims 1-12. No new matter is presented. Thus, claims 1-12 are pending and under consideration. The rejections are traversed below.

REJECTION UNDER 35 U.S.C. §103(a):

Claims 1-12 were rejected under 35 U.S.C. § 103(a) as being unpatentable over various combinations of the following: U.S. Patent No. 5,684,955 (Meyer), U.S. Patent No. 6,438,746 (Martin) and U.S. Patent No. 5,452,461 (Umekita).

The Examiner refers to col. 9, lines 38-46 of Meyer that reads, "stub objects should behave towards the other objects in exactly the same way as real objects" and indicates that the stub objects and the real objects have the same behavior. However, if the stub objects and the real objects had the same behavior, they would not be separately generated. When interpreting the portion cited by the Examiner as a whole, it indicates that other objects can interact with the sub object as if it is a real object, where the stub object cannot execute the same processing as the real object, which is apparent from the description at col. 7, lines 20 through 61, FIG. 5 and corresponding text).

Col. 9, lines 38-46 of Meyer specifically states:

> "representative objects should be present as the stub objects instead of the real objects in a distributed application in operating system processes which do not themselves contain a real object" and "said stub objects are then available locally as a contact point in the event of a request to the real object..."

As can be seen from the above discussion, the stub object in Meyer does not have the same function as the real object. For the above-discussed reason, the Examiner does not appear to have established a prima facie case of obviousness.

In contrast, according to the present invention, because parallel processing should be carried out, the same objects are instantiated for the parallel processing. On the other hand, in Meyer, because the distribution of the objects is carried out, the representative object, which is the stub object of the real object that is remotely instantiated, is necessary in the local computer. Therefore, the different function from the real object is realized in the stub object. This is clear from the lack of description related to "the class" in the cited portions in Meyer.

Furthermore, SX_NEW instantiates a real object in another operating system process and a stub object in a local computer. As described above, because the stub object does not correspond to "an object of the class", "an instruction to call a construction instruction routine for an object of the class" is not generated in Meyer. The same applies to the SX_DELETE.

In addition, as cited by the examiner, col. 10, lines 14-17, "the calls NEW and DELETE are also replaced in the source by the preprocessor by the function SX_NEWobj as instantiation function and SX_NEW as delete function." However, "an instruction to call a construction instruction routine for an object of the class" is generated "before said execution statement to be executed in parallel or an execution statement to be parallelized by said parallelization directive". In addition, "an instruction to call a destruction instruction routine for the generated object of the class" is generated "after said execution statement to be executed in parallel or said execution statement to be parallelized by said parallelization directive" (see for example, claim 1).

As described in col. 10, lines 47-58 of Meyer, for example, the protocol information file contains "parameter string", which is "used for packing and unpacking the parameters." This packing and unpacking means "V642: Pack parameters" and "V650: Unpack parameters" in FIG. 5 shows a method call included in the process step V600: Execution (see, FIG. 5 and corresponding text). Therefore, this protocol information file is used in the execution stage. However, the intermediate language is normally not used in the execution stage, but is used in the compiling stage. Therefore, the protocol information file is not an intermediate language.

The Examiner acknowledges that Meyer does not disclose that the execution statement is executed in parallel or is parallelized as specified in the parallelization directive, but relies on Martin as teaching the same. Martin is directed to entering functional requirements on system performance criteria as comment fields in an object-oriented language, which are ignored by a conventional compiler to allow compilation of a single processor version of the program, and are interpreted by a pre-compiler to take into account system data when compiling code for host computers of the distributed system (see, col. 2, lines 25-29 and col. 9, lines 44-54).

Independent claim 1, by way of example, recites, "detecting that a certain class-type variable is contained in an execution statement to be executed in parallel or a certain class-type variable is specified in a parallelization directive as a class to be parallelized." Claim 1 further recites generating "an instruction to call a construction instruction routine for an object of the class before said execution statement to be executed in parallel or an execution statement to be parallelized by said parallelization directive." The claimed invention in claim 1 includes

generating "an instruction to call a destruction instruction routine for the generated object of the class after said execution statement... or said execution statement... to destruct the generated object in addition to said original object of the class." Independent claims 5 and 9 also recite similar features.

The cited references, do not teach or suggest, "detecting whether a certain class-type variable is contained or a certain class-type variable is specified", "generating an instruction to call a construction instruction before said execution statement or an execution statement to be parallelized" and "generating an instruction to call a destruction instruction after said execution statement or said execution statement", as recited in claim 1 (see also claims 5 and 9).

The cited references, alone or in combination, do not teach or suggest the above-discussed features of independent claims 1, 5 and 9.

The Examiner further combines Umekita with Martin and Meyer to reject dependent claims 4, 8 and 12. According to Umekita, intermediate codes of a source program are divided into tasks and sequential relationships among the tasks are generated for sequentially distributing the tasks based on a total processing time of processors (see, FIG. 27 and corresponding text).

Dependent claims 4, 8 and 12 recite that the compiler is for "a parallel computer with shared memory." Umekita does not teach or suggest the claimed compiling method and system generating instructions for parallel processing of a source program where the compiler is for "a parallel computer with shared memory" (claims 4, 8 and 12 including respective independent claims 1, 5 and 9).

It is submitted that the independent claims are patentable over the cited references.

For at least the above-mentioned reasons, claims depending from the independent claims are patentably distinguishable over the cited references. The dependent claims are also independently patentable. For example, claim 2 recites, "allocating a construction and destruction instruction information region in the intermediate language of the class, when a class-type variable which has possibility to be executed in parallel is specified" when generating an intermediate language from the source program. The cited references, alone or in combination, do not teach or suggest these features of claim 2.

Therefore, withdrawal of the rejection is respectfully requested.

REQUEST FOR WITHDRWAL OF FINALITY OF THE OFFICE ACTION:

In light of the above arguments, Applicants respectfully submit that the features recited in each of the independent claims are patentably distinguishable over the cited references.

Therefore, withdrawal of the finality of the Office Action is respectfully requested.

CONCLUSION:

There being no further outstanding objections or rejections, it is submitted that the application is in condition for allowance. An early action to that effect is courteously solicited.

Finally, if there are any formal matters remaining after this response, the Examiner is requested to telephone the undersigned to attend to these matters.

If there are any additional fees associated with filing of this Amendment, please charge the same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: 03/29/2006

By: Temnit Afework

Temnit Afework
Registration No. 58,202

1201 New York Avenue, NW, Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501